



## Strathprints Institutional Repository

**Lopez, Alejandro and Nicholls, William and Stickland, Matthew and Dempster, William (2015) CFD study of jet impingement test erosion using Ansys Fluent ® and OpenFoam ®. Computer Physics Communications, 197. pp. 88-95. ISSN 0010-4655 , <http://dx.doi.org/10.1016/j.cpc.2015.07.016>**

This version is available at <http://strathprints.strath.ac.uk/54133/>

**Strathprints** is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<http://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: [strathprints@strath.ac.uk](mailto:strathprints@strath.ac.uk)

# CFD study of Jet Impingement Test erosion using Ansys Fluent® and OpenFOAM®

Alejandro López<sup>a,\*</sup>, William Nicholls<sup>a</sup>, Matthew T. Stickland<sup>a</sup>, William M. Dempster<sup>a</sup>

<sup>a</sup>*Department of Mechanical and Aerospace Engineering, Strathclyde University, Glasgow,  
United Kingdom*

---

## Abstract

The initial aim of this study was to compare OpenFoam and Ansys Fluent in order to verify OpenFoam's Lagrangian Library and erosion capabilities. However, it was found that previous versions of Fluent have been providing wrong results for the discrete phase and the differences with the latest version (Ansys Fluent 15) are shown. A Submerged Jet Impingement Test is an effective method for studying erosion created by solid particles entrained in a liquid. When considering low particle concentrations a Lagrangian modeling of the particulate phase is a reasonable approach. Proper linkage between OpenFOAM's Lagrangian library and the solver pimpleFoam for incompressible transient flows allows two-phase simulations to be undertaken for comparison with Ansys Fluent with the aim of verifying OpenFoam's accuracy. Steady state convergence for the fluid flow is first accomplished and the results are compared, confirming a good agreement between the two packages. A transient simulation was then set up and spherical particles incorporated into the fluid flow. An assessment of the two codes' discrete phase models was carried out, focusing on the differences between impact angles and velocities yielded at the impingement plate's surface employing a similar strategy to that outlined first by Hattori et al [1] and later by Gnanavelu et al. in [2, 3]. In the comparison of OpenFoam with the latest version of Fluent, the main differences between the injection models

---

are highlighted and the coupling possibilities between phases are taken into consideration. Agreement between trends for both impact angles and velocities is satisfactory when the last version of the commercial package is considered and the average discrepancy between numeric values is very low, verifying OpenFoam's Lagrangian library. Two different Jet Impingement Test configurations are also compared and the differences highlighted.

*Keywords:* Fluent, OpenFOAM, multiphase, discrete phase model, erosion, Jet Impingement Test

---

## 1. Introduction

Erosion by solid particle impingement is a very common process in many applications. For over half a century, a very large number of papers have been published on the matter. However, there seems to be very little agreement with respect to its numerical prediction. Meng and Ludema [4] carried out a broad literature review of more than 5000 papers dating from 1957 to 1992. In their article they separated 28 erosion models out of the almost 2000 existing empirical models, a fact which exemplifies the poor agreement between authors on this subject. One of the few theories on which there seems to be some kind of agreement describes two mechanisms acting together to produce the wear scar: cutting and deformation wear. When particles hit the surface and they tear material away with them in a cutting action, it is called cutting wear. This mechanism is the predominant one for ductile materials and particles impinging at low angles of attack with respect to the surface being eroded. Alternatively, several particles might impact on the same place transferring some of their kinetic energy to the surface in the form of hardening work [5]. According to this theory, in a given collision with the target material, as soon as the particle contacts the surface, stress concentrations appear as a result of the elastic deformation that takes place. If these stresses are not over the elastic limit of the target material, and also leaving aside fatigue damage effects, they should cause no deformation. However, if the elastic limit is reached, plastic

deformation will occur at the location of the maximum stress. The repeated impacts create then a plastically deformed layer that will deform further upon repetition of particle collisions. This deformation causes hardening and increases the elastic limit in that region turning the material harder and more brittle, reaching a point when it can no longer be plastically deformed. Eventually, upon further load, pieces of the material's surface separate from the target and are carried away by the fluid. This hypothesis was studied by Davies in [6], and Van Riemsdijk and Bitter in [7] and then adopted by several authors [8, 5, 9, 10, 11]. This mechanism is called deformation wear and it predominates for high angles of impingement and in brittle materials. In a more recent study carried out by Gnanavelu et al [2, 3], a new methodology based upon a combination of both an experimental submerged Jet Impingement Test (JIT) on a circular target and Computational Fluid Dynamics (CFD) has been developed. This procedure yields a material specific relationship for calculation of the wear rates. In this article, thorough comparison of a Jet Impingement Test simulated with Ansys Fluent (Fluent) 15.0 and OpenFOAM 2.2.x. (OpenFoam) is presented in order to verify that OpenFoam's combination of libraries is able to reproduce correctly the particle behavior in the Jet Impingement test, focusing on the correlation between particle variables and the radial distance from the centre of the target. Implementation of several erosion models and testing all of them simultaneously is of great interest and OpenFoam proves itself, after comparison, as an asset for erosion modeling. Two different configurations of the JIT were set up and compared, being those implemented by Hattori et al. in [1] and Gnanavelu et al. in [2, 3]. However, when the same simulation corresponding to the 5 mm Jet Impingement Test was set up in Fluent 14 and OpenFoam 2.2.x, huge differences in both particle velocities and angles were found. OpenFoam's code was thoroughly checked and it's validity confirmed, while, as the latest Fluent version (15) was available, it was verified that due to a programming error that has been corrected in that latest update, previous versions of the commercial software had been providing wrong values for the particles. These results are discussed here and some of the articles affected by these miscalculation are

reviewed. The CFD results confirm that the maximum and minimum velocity magnitude of the impinging particles does not experience large variations when the distance between nozzle exit and target is increased to 25 mm. However, the effect is more visible in its profile variation along the target's radius.

## 2. The Euler-Lagrange Approach

When dealing with particles entrained in a fluid flow there are a number of possibilities available for the numerical treatment of the dispersed phase. The number of operations needed for the calculation of the particle trajectories may become numerically unmanageable if high concentrations are taken into account. In these cases, treatment of the dispersed phase as a second Eulerian phase is common practice. As the concentration diminishes, solids may be handled in a Lagrangian way, defining the forces acting on the discrete phase and obtaining velocities and positions by means of integration of Newton's equations (1, 2).

$$m_p \frac{d\vec{u}_p}{dt} = \vec{F}_p \quad (1)$$

$$\frac{d\vec{x}_p}{dt} = \vec{u}_p \quad (2)$$

Where  $m_p$  and  $\vec{u}_p$  are the mass and velocity of the particle,  $\vec{F}_p$  is the term corresponding to the forces acting on the particle and  $\vec{x}_p$  is the position of the particle. The Lagrangian approach may be subdivided by the kind of coupling between particles and fluid:

- One-way coupling: If the concentration of particles is low enough, as in the case discussed here, the influence that the particles exert on the fluid phase may be neglected.
- Two-way coupling: In this case the force exerted by the particles is no longer neglected and a new term should be included into the fluid's equations in order to account for this.

- Four-way coupling: In this case also particle-particle interactions are taken into account.

Furthermore, the discrete phase can be modeled as computational parcels, which are groups of particles with average quantities for the main variables, or single particles. In the case taken into consideration in this article, one-way coupling of single particles is selected in both Fluent and OpenFoam due to the very low concentration of solids, which are in the range of 1% by volume and the particles are only affected by the drag force, as other forces like gravity or pressure gradient are negligible. In this case, the momentum transfer from the particles to the fluid phase can be neglected, and this was confirmed by computing the momentum transfer in two-way coupled simulations with the same amount of solids.

### *2.1. Lagrangian and Eulerian phases coupling in OpenFoam*

Fluent gives the user the possibility of adding a cloud of particles and tracking their movement at any given point. However, in order to do the same in OpenFoam, some programming was required, because of the non-existence of an Eulerian-Lagrangian solver amongst the default ones in version 2.2.x. To do this, a linkage between an incompressible transient solver and the Lagrangian intermediate library must be implemented. The incompressible solver chosen was `pimpleFoam` which, as its name states, has the merged PISO-SIMPLE algorithm implementation. The linkage was established through the inclusion in the top level solver of the header file responsible for the definitions of the particulate phase and a call to the function that uses the fluid flow variables to calculate the forces acting on the solids for a later integration to obtain their velocities and positions. All the properties of the particles, as well as the templates used to gather the necessary variables at impingement, are defined through a dictionary which is read when the simulation starts.

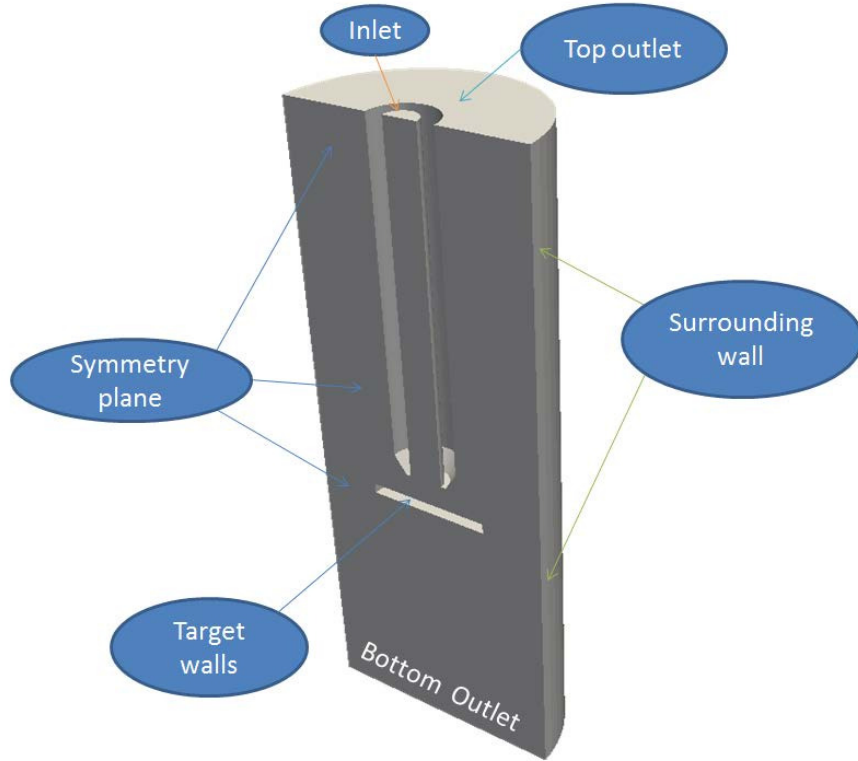
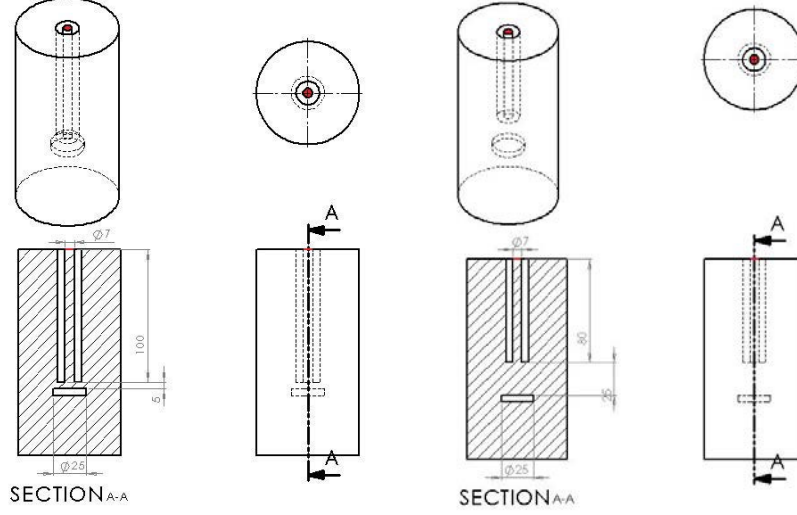


Figure 1: 3D geometry used showing the location of the boundary conditions

### 3. Simulation parameters

Two different configurations of the normal jet impingement test were implemented at distances of 5 mm and 25 mm. Figure 1 shows the location of the boundary conditions. Schematics of both configurations are shown in Figures 2a and 2b. The velocity magnitude at the inlet of both was chosen to be uniform with a value of  $10 \frac{m}{s}$ . For each of the cases, one symmetry plane was used so only half of the geometry was simulated and the mesh used was the same one for OpenFoam and Fluent.

Once the steady-state convergence was reached for the continuous phase in



(a) 5mm JIT domain with the inlet boundary highlighted in red      (b) 25 mm JIT domain with the inlet boundary highlighted in red

both Fluent and OpenFoam, a uniform distribution of  $250 \mu m$  spherical particles were incorporated into the flow and their impacts against the target were individually monitored. The domains shown in Figures 2a and 2b were discretized into 984960 and 670349 cells respectively, paying special attention to the  $y^+$  around the target which was considered adequate for the K-Epsilon turbulence model used. Two different configurations of the normal jet impingement test were implemented at distances from the pipe outlet and to the target of 5 mm and 25 mm.

#### 4. Eulerian phase steady-state

Convergence criteria were satisfied for the fluid phase when the residuals fell below  $10^{-4}$  and this was achieved by both packages in a similar amount of iterations for both the 5 mm and the 25 mm cases. The SIMPLE algorithm was chosen for the pressure/velocity coupling and the same set of discretization schemes, solvers and boundary conditions were used in both simulations. The inlet was also situated sufficiently upstream to allow the fluid flow to develop



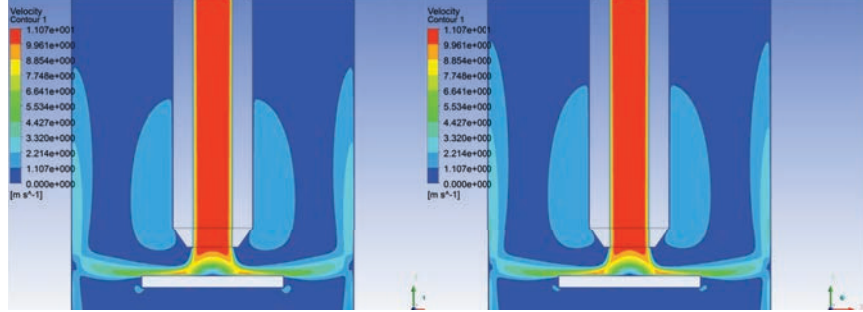


Figure 3: 5 mm separation Steady state comparison. OpenFOAM(right) and Ansys Fluent (Left)

inside the pipe before it's outlet and right before impinging on the target. As shown in Figures 3 and 4, minor differences can be observed in the steady state flow fields from both CFD packages. The largest discrepancies between solutions are located at the corners of the jet as shown in Figures 5a and 5b, which are contour plots of the absolute difference between the results of each package. In the case of the 5mm distance the highest value of the relative difference is 6%, affecting a very small region of the domain. In the 25mm case, the maximum relative difference is 9.7% with a similar region affected. These discrepancies may be attributed to some small differences in the calculation algorithms. The differences between velocities in the region of the domain where the impact velocities and angles of the particles at the moment they reach the test target are considered, and which is most significant for this study, are minor and the fields can be considered to be identical.

## 5. Discrete phase modeling

Once the steady-state values for the main flow field variables were obtained, these were set as the initial conditions for the transient simulations. In fact, the difference between a transient simulation with one or two iterations for the Eulerian phase and another one in which no iterations were considered yielded the same values for the variables of the particles. Particle tracking inside the

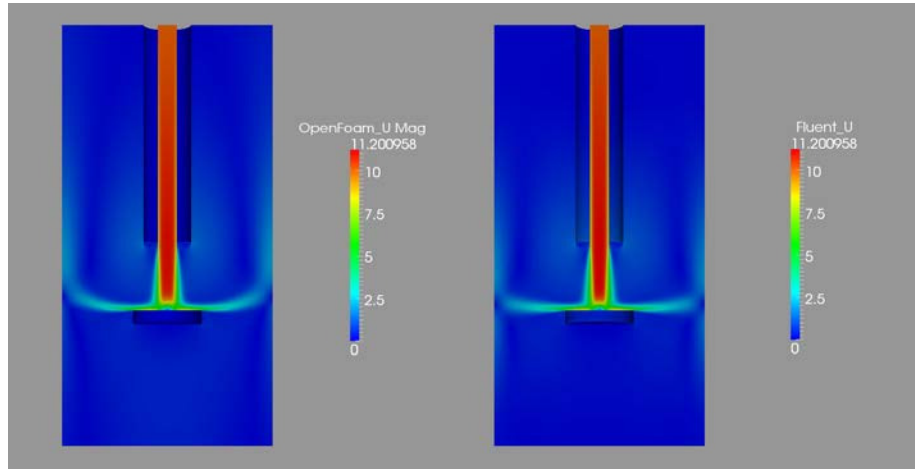
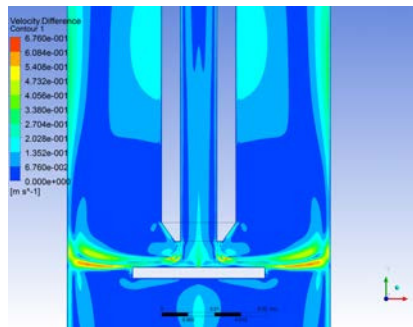
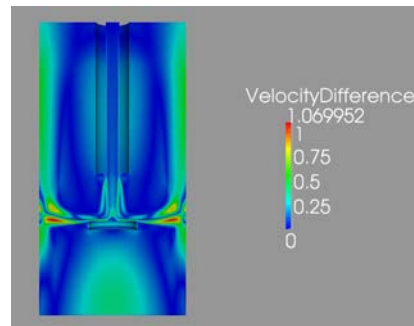


Figure 4: 25 mm separation Steady state comparison. Units in m/s



(a) Contours of the absolute differences for the 5 mm separation case



(b) Contours of the absolute differences for the 25 mm separation case. Units in m/s

fluid was carried out in a similar way in both packages. The user needs first to define which are the most relevant forces influencing the particle. In Fluent, the drag force was chosen as the sole force acting on the particles and the same arrangement was made in OpenFoam. The rest of the forces, including gravity, were ignored in this simulation due to their negligible influence on the discrete phase in comparison with the drag force. The force balance according to equation 1 is shown in equation 3:

$$F_p = m_p \frac{du_p}{dt} = F_D \quad (3)$$

The drag force on spherical particles takes the form of equation 4 and the drag coefficient in OpenFoam is obtained from equation 5, while in Fluent, the formula developed by Morsi and Alexander in [12] is employed. The drag formula in [12] was also implemented and tested in OpenFoam having found no differences in what respects to the impingement conditions.

$$F_D = m_p \frac{18\mu}{\rho_p d_p^2} \frac{C_D Re_p}{24} (u - u_p) \quad (4)$$

$$C_D = \begin{cases} \frac{24}{Re_p} & : Re_p < 1 \\ \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) & : 1 \leq Re_p \leq 1000 \\ 0.44 & : Re_p > 1000 \end{cases} \quad (5)$$

### 5.1. Particle Injection

Differences arose when taking into consideration the injection models available in both software packages. In order to inject the same mass of particles, surface injection was chosen in OpenFoam and single injections in Fluent. One of the main differences regarding the injection models is that, even though both softwares use the name *surface* injection, they differ a lot from each other. OpenFoam's surface injection calculates the number of particles to be injected per time-step and, using a random function, chooses different locations at the selected surface for the different injections. In Fluent however, when surface injection is selected, the number of particles injected per time-step will be equal

to the number of faces on the selected surface. For this reason, two single injections are chosen in two different locations at the inlet, so that the number of particles released matches that in OpenFoam. If surface injection had been used in Fluent, the number of particles would have been much greater. Despite OpenFoam’s randomization of the particle injection, if stochastic dispersion is not included, impact locations on the target will lack randomness. Fluent faces the same lack of randomness problem, given that all the particles are released from the same point. In this case, the so called, discrete random walk model was selected in Fluent. In the case of OpenFoam, the stochastic dispersion is included through the *kinematicCloudProperties* dictionary. The particles chosen were spheres with a uniform radius distribution of  $250\ \mu\text{m}$ , density of  $2206\ \frac{\text{Kg}}{\text{m}^3}$  in the 5 mm case and  $2300\ \frac{\text{Kg}}{\text{m}^3}$  in the 25 mm one together with an initial velocity equal to the fluid’s inlet velocity ( $10\ \frac{\text{m}}{\text{s}}$ ). A thorough study of the particle trajectories revealed that modification of the particle inlet velocity (setting it to  $0\ \frac{\text{m}}{\text{s}}$  for instance) was not perceived downstream due to the small distance the particles travelled to accelerate close to the fluid’s velocity and the comparably long distance to the target’s surface.

## 5.2. Particle variables gathering

In Fluent, a User Defined Function (UDF) was created which gathered the particle velocity, particle angle of impingement and impact location in terms of radius along the target. In addition to this, the UDF also terminates the particle trajectory once they impact the target, so that no second impacts of the same particle were possibly considered. The same condition was met in OpenFoam. However, in this case, this was implemented via the *kinematicCloudProperties* dictionary, defining an *escape* type for the target’s boundary. OpenFoam’s solution for the output of particle variables was achieved through a *cloudFunctionObject* called *patchPostProcessing*. This template allows printing of the desired particle variables into a text file so that radius, velocity and angle of impingement were successfully gathered, calculated and stored as soon as the particles struck the surface.

### 5.3. Transient simulation features

Some of the features corresponding to the transient simulation are shown in Table 1:

Variable	Units	Value
Time-step	s	1.8759e-05
Number of time-steps	-	53308
Particle material	-	Carbon
Particle diameter distribution	-	Uniform
Particle diameter value	m	2.5e-04
Coupling between phases	-	One-way
Forces	-	Drag
Drag coefficient	-	Sphere drag
Particle density	$\frac{Kg}{m^3}$	2206

Table 1: Transient simulation features

When considering a volume fraction of the solids of 1%, the yielded mass flow rate of solid particles was as low as  $0.000962 \frac{kg}{s}$ , which in turn, yielded a rate of 53308 particles per second. This allowed setting up the time-step for the transient simulation as  $1.8759 * 10^{-5}$  seconds. During the simulation in OpenFOAM the Courant number was monitored also verifying that the particles did not travel too fast through the mesh cells and their trajectory was accurately calculated.

## 6. Impingement conditions

Once all the impingement variables of the more than 50.000 particles were gathered, they were post-processed in order to obtain their average compared to the distance from the centre of the circular target. In order to do this, the 25 mm diameter circle was divided into smaller concentric regions separated by 1 mm from each other as represented in Figure 6. The impacts located within each

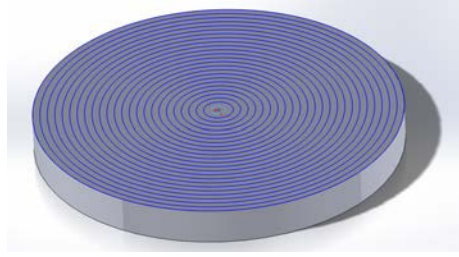
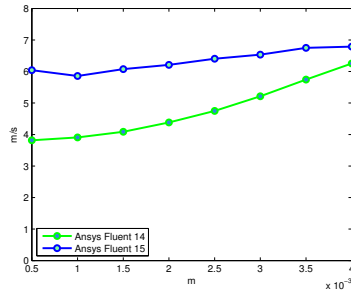
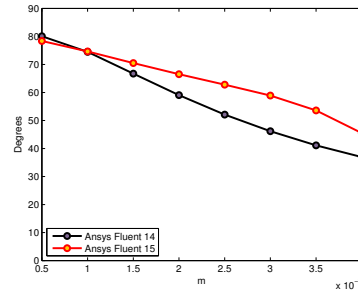


Figure 6: 25 mm diameter cillindrical target subdivided into 1mm regions



(a) Average particle velocities ( $\frac{m}{s}$ ) at impingement versus distance from the centre of the target ( $m$ ) in the two Fluent versions



(b) Average impact angles at impingement (*degrees*) versus distance from the centre of the target ( $m$ ) in the two Fluent versions

of these regions were then averaged and a mean value for both the angle and the velocity was obtained for each of the software packages and compared with each other. A surprising fact was that, for the same parameters, different versions of Ansys Fluent (namely 14.0 and 15.0) yielded very different particle velocity profiles, which may indicate that a programming irregularity was present in previous versions but has been corrected in the latest one, thus affecting a wide range of publications. This would have been impossible to detect unless a comparison with other software packages like the present one had been carried out. Velocities presented here correspond to Fluent 15.0, while in Gnanavelu et al. [2, 3], a previous version of Fluent (dating from 2011) was used in which the velocity profiles deviated from their correct values due to the irregularity. However, this should not affect the methodology but only the results concerning those simulations and the graphs obtained from them. In fact, the use of data from the corrected version of Fluent should yield results that fit much better to the experimental data than the existing ones. Figures 7a and 7b show the differences in the particle velocity and angle of impingement profiles between two cases which were exactly the same but solved by the two versions of Fluent when the separation between nozzle outlet and target was 5 mm. These results can be confirmed by comparing them to the profiles obtained by Gnanavelu et al in [2] with Fluent and those obtained by Hattori et al in [1] with the *Star-CD* commercial code.

Figure 9 illustrates the location of all the particle impacts. Most of these were located within the 0-4 mm region of the target when the separation between the nozzle and the target was 5 mm. As the number of impacts in the outer region was not high enough, values of the average in those regions were not considered due to the low density of these in comparison with the impact density in the inner (0-4 mm) region.

### 6.1. Velocity at impingement versus radius

As shown in Figures 10a and 10b, minor differences can be observed between the profiles of the particle velocity along the radius for OpenFoam and Fluent

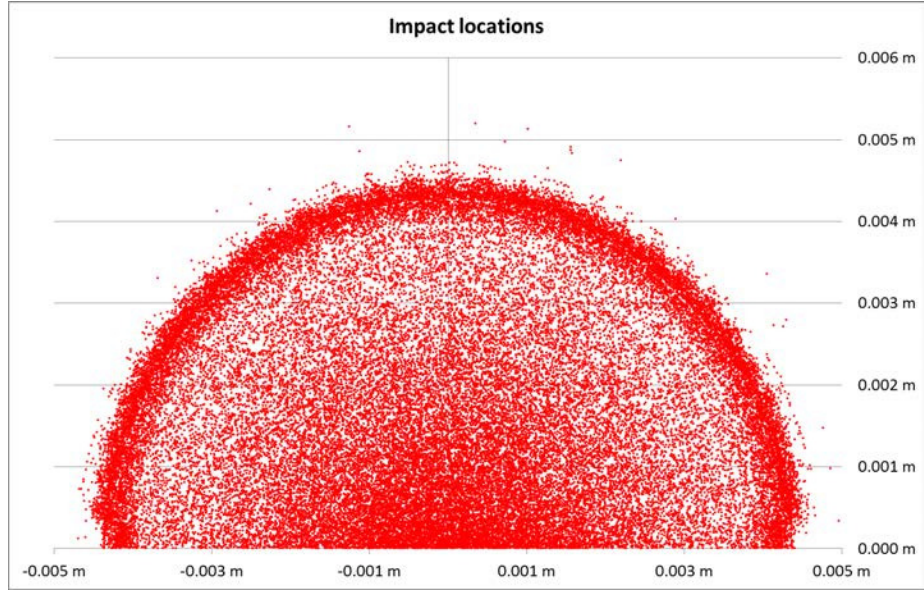


Figure 8: Impact locations for the 5mm nozzle distance case obtained with OpenFoam

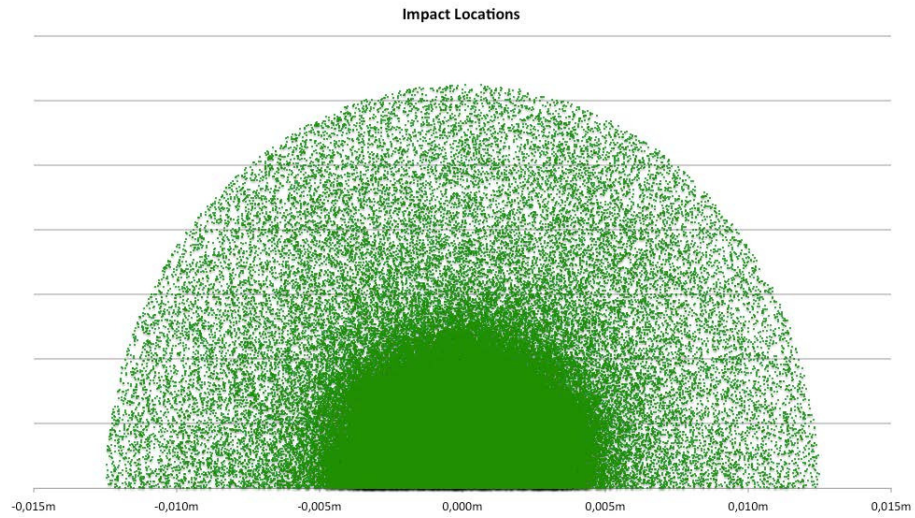
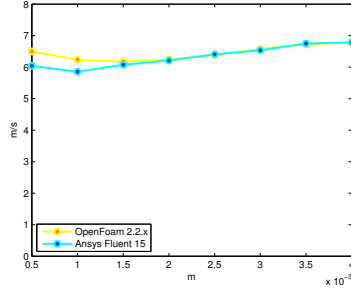
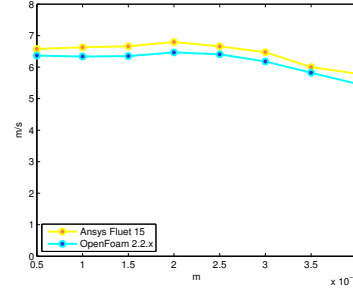


Figure 9: Impact locations for the 25mm nozzle distance case obtained with OpenFoam





(a) Fluent 15.0 Vs OpenFOAM 2.2.x  
particle velocities at impingement ( $\frac{m}{s}$ )  
for the 5 mm distance case



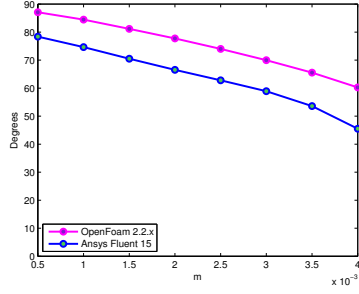
(b) Fluent 15.0 Vs OpenFOAM 2.2.x  
particle velocities at impingement ( $\frac{m}{s}$ )  
for the 25 mm distance case

15, with a mean relative difference of 2%, which represents more than a 26% improvement if compared to Fluent 14. It is worth noting that, in the 5mm case, impingement velocities at the centre of the target have lower values, being these values higher the further away from the centre. When the separation between nozzle and target is bigger, the contrary is true, being the velocities at the centre slightly higher than further away from it.

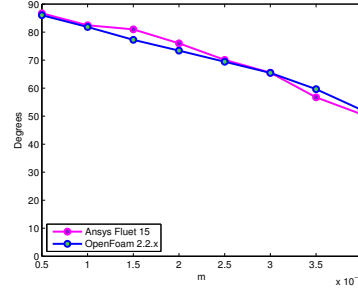
This is probably due to the disparity in the accelerations that both fluid and particles experience in the two different gaps left between the target and the nozzle exit.

## 6.2. Angle of impingement versus radius

Figures 11a and 11b represent the trends followed by the angles of impingement for both software packages under the two different configurations of the JIT. Improved agreement between both Fluent and OpenFoam is once more corroborated, being the average relative difference of 15.3%, which is a 10% improvement if compared with the Fluent 14 results (25.2%). The small differences could be attributed to the slightly different steady state results, in which differences in the velocity close to the wall would account for the differences in the angles of impingement. However, both softwares capture the same slope for the behavior of the angle of impingement along the target's radius. In this case,



(a) Fluent 15.0 Vs OpenFOAM 2.2.x angles at impingement (*degrees*) for the 5 mm distance case



(b) Fluent 15.0 Vs OpenFOAM 2.2.x angles at impingement (*degrees*) for the 25 mm distance case

the difference in the slope of the lines formed by the averaged angles of impingement is smaller in comparison with that of the velocities. Results indicate that this slope is slightly steeper for the 25 mm distance JIT.

## 7. Erosion modeling in OpenFOAM

Erosion is a very complex process, involving a very large number of variables. There is a significant number of papers on erosion, containing many different approaches on how to predict it. Table 2 shows some of the factors that have been taken into consideration in the literature divided in three blocks corresponding to those affecting the particles, the surfaces involved in the process or the carrier fluid.

For particles		For surfaces		For the carrier fluid	
1	Impact and rebound angles	1	Physical properties	1	State of motion (laminar versus turbulent)
2	Impact and rebound speeds	2	Change in shape caused by erosion	2	Velocity
3	Rotation before and after impact	3	Stress level	3	Temperature
4	Shape and size	4	Temperature	4	Chemical composition and physical properties
5	Volume concentration and surface flux	5	Presence of oxide (or other) coatings		
6	Physical properties (hardness strength and density)	6	Simultaneous occurrence of corrosion		
7	Fragmentation				
8	Interactions (with surfaces, fluid or other particles)				
9	Temperature				
10	Presence of additives				
11	Electrical charge				

Table 2: Factors affecting erosion

Existence of such a high number of variables and ways to predict erosion results in the following procedure being common practice. Most of the times, performance of different approaches and formulae on the real case are tested and, from the results, the model that suits best the recorded wear scar is chosen for forecasting similar cases. This is all based on the assumption that the conditions in the CFD simulation are close enough to those on the real problem. The results found in this article show that, CFD can sometimes be misleading if there is a lack of validation data for the given algorithm. The structure of OpenFoam's Lagrangian library is based on C++ templates. This structure allows, in the context of erosion, defining as many additional models as required for simultaneous calculation of erosion, making the task of comparing the performance of several approaches for a single problem sensibly easier. Each particle impinging the surface generates a value for erosion at the cel being hit. All the impacts are summed up and stored in a field of scalars with a value for each face of the selected boundary. By generating several erosion templates and calling them during runtime, the possibility of having erosion calculated by different formulae in the same simulation is enabled.

## 8. Conclusions

Minor discrepancies were found between Fluent 15 and OpenFoam, considering the absolute difference in the steady-state simulations and the portion of the domain affected by these, it can be said that both results are basically equivalent, verifying OpenFoam's accuracy and the validity of the package for steady-state calculations of a single incompressible phase as well as for multi-phase simulations of a Lagrangian phase dragged by a continuous phase. In fact, versions of Fluent since, at least, 2011 have been providing miscalculated values of the particle variables due to, most likely, a small programming instability in the discrete phase model that is responsible for integrating the sum of the forces divided by the mass in order to obtain velocities and positions. Unfortunately, this may affect some of the published literature including some of the references

mentioned in this paper [2, 3]. With respect to the discrete phase model of the latest Fluent version, also minor differences were found. The level of agreement between trends for the particle velocities and angles of impingement in both software packages is very good. Considering the two different configurations, as the figures 7a and 7b illustrate, for the same values of the inlet particle and fluid velocities, the profiles yielded, especially the one representing the particle velocity, are visibly different. Two of the differentiating factors lie in the location and concentration of the impacts and in the trends for the velocity and the angle of impingement. As the particles have more space to travel between the nozzle outlet and the impact surface, the impacts are more distributed over the whole target, which makes the impact rate lower, given that more particles will leave the area without an impact. Steeper slopes are obtained for velocity and angle of impingement in the 5 mm case, due to the greater bend the fluid has to overcome in the smaller gap left for it to flow between the nozzle exit and the target's surface. It is this bend which induces the different slopes in the average velocities between the two cases considered, showing an increasing velocity as the distance from the centre is increased in the 5mm case and a decreasing one for the 25mm case. Finally, the next phase to be completed is the finishing of the experimental validation for this model, being concerned at the moment with the design of an appropriate test rig for this purpose, which, combined with mesh deformation techniques, will allow to determine how erosion develops taking into account the surface deformation as it occurs.

## 9. Acknowledgement

Results were obtained using the EPSRC funded ARCHIE-WeSt High Performance Computer ([www.archie-west.ac.uk](http://www.archie-west.ac.uk)). EPSRC grant no. EP/K000586/1.

## References

- [1] S. H. Kenichi Sugiyama, Kenji Harada, Influence of impact angle of solid particles on erosion by slurry jet, *Wear* 265 (2008) 713–20.

- [2] A. G. et al., An integrated methodology for predicting material wear rates due to erosion, *Wear* 267 (2009) 1935–44.
- [3] A. G. et al., An investigation of a geometry independent integrated method to predict erosion rates in slurry erosion, *Wear* 271 (2011) 712–9.
- [4] K. C. L. H. C. Meng, Wear models and predictive equations: their form and content, *Wear* 181 (1995) 443–57.
- [5] I. M. Hutchings, A model for the erosion of metals by spherical particles at normal incidence, *Wear* 70 (1981) 269–81.
- [6] R. M. Davies, The determination of static and dynamic yield stresses using a step ball, *Proceedings of the Royal Society of London* 197A (1949) 416–432.
- [7] J. G. A. B. A.J. Van Riemsdijk, Erosion in gas-solid systems, Fifth World Petroleum Congress, Section VII, Engineering, Equipment and Materials (1959) 43–58.
- [8] G. G., T. W., An experimental investigation of the erosion characteristics of 2024 aluminum alloy, Department of Aerospace Engineering Tech. Rep., University of Cincinnati (1973) 73–7.
- [9] J. G. A. Bitter, A study of erosion phenomena Part I, *Wear* 6 (1963) 5–21.
- [10] J. G. A. Bitter, A study of erosion phenomena Part II, *Wear* 6 (1963) 169–90.
- [11] J. H. Neilson, A. Gilchrist, Erosion by a stream of solid particles, *Wear* 11 (1968) 111–22.
- [12] S. Morsi, A. Alexander, An investigation of particle trajectories in two-phase flow systems, *Journal of Fluid Mechanics* 55(2) (1972) 193–208.